PCIe® 4.0 PHY Logical

Debendra Das Sharma, PhD

PHY Logical Sub-Team Lead

Senior Principal Engineer and Director of I/O Technology and Standards Data Center Group

Intel Corporation

Disclaimer

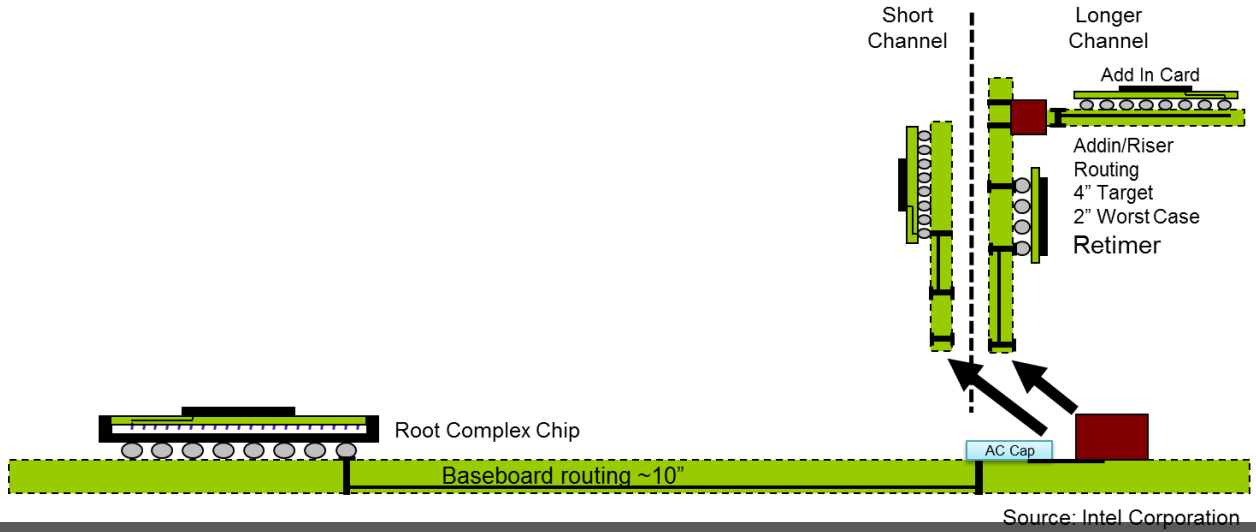
The information in this presentation refers to specifications still in the development process. This presentation reflects the current thinking of various PCI-SIG® workgroups, but all material is subject to change before the specifications are released.

Agenda

- **Background**
- **128b/130b Encoding Scheme**
- **Transmitter Equalization and Training**
- **Testability Features and Receiver Margining**
- **Configuration Registers for 16.0 GT/s**
- **Summary**

PCIe 4.0 Background

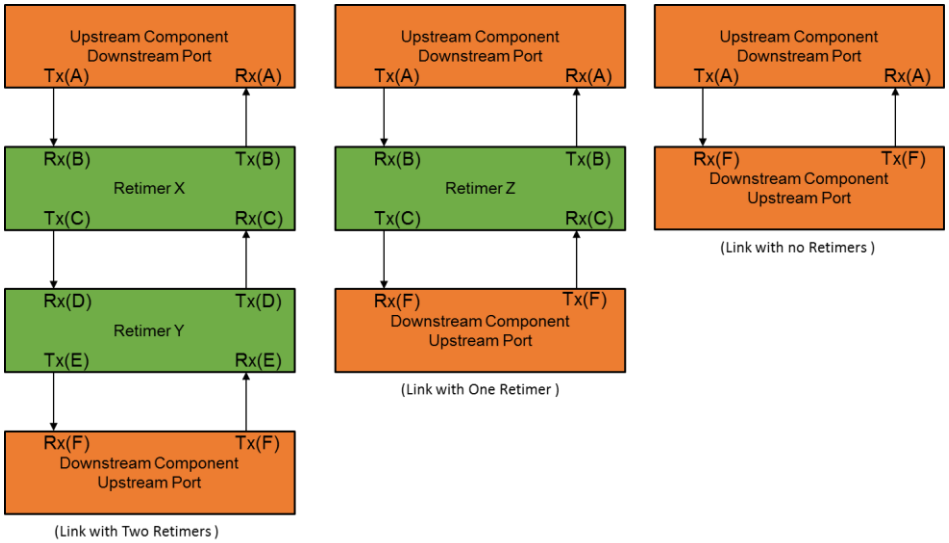
- **PCIe® 4.0 data rate: 16.0 GT/s**
  - High Volume Manufacturing channel for client/ servers
  - Low power and ease of design
  - Fully backwards compatible with PCIe 3.0 (8.0 GT/s), PCIe 2.0 (5.0 GT/s) and PCIe 1.0 (2.5 GT/s). Doubling per-pin B/W every generation!
- **Connector improvements to reduce cross-talk and improve insertion loss at 8G Nyquist**
- **PCIe 4.0 Channel insertion loss budget 28 dB. 2 Connector 20" server PCIe topology needs either re-timer or ultra low-loss PCB**



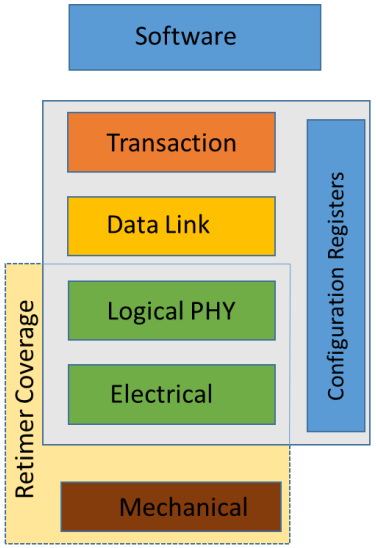
Source: Intel Corporation

Retimers for Extending Channel Reach

- **Channel Extension devices; up to 2 Retimers**
- **Part of PCIe base spec (3.1+)**
- **Critical for longer server channels in PCIe 4.0**
- **Has the Electrical and PHY Logical – no Link/ Transaction layer, no config registers, no in-band access by S/W (PCIe 4.0 allows optional read access through a new Ordered Set)**
- **Actively participates in link training, power management, clock compensation, Link Equalization**
- **Electrically separate links on either end of Re-timer**
- **8b/10b TS2'es enhanced to report "*Retimer Present*" and "*Two Retimers present*" during *Configuration.Complete* state which will be reflected in *Link Capabilities 2* register**



(Various System Topologies with or without Retimers)



(Source: Intel Corporation)

Agenda

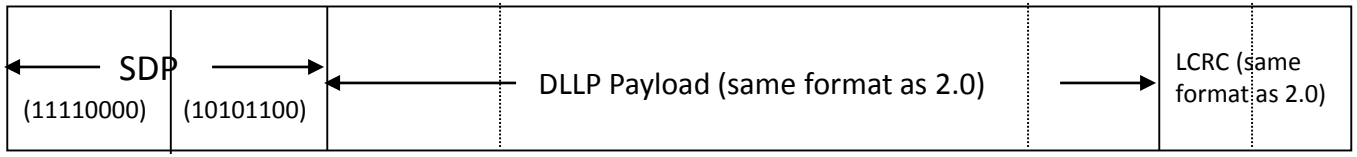
- Background
- **128b/130b Encoding Scheme**
- Transmitter Equalization and Training
- Testability Features and Receiver Margining
- Configuration Registers for 16.0 GT/s
- Summary

128b/130b Encoding

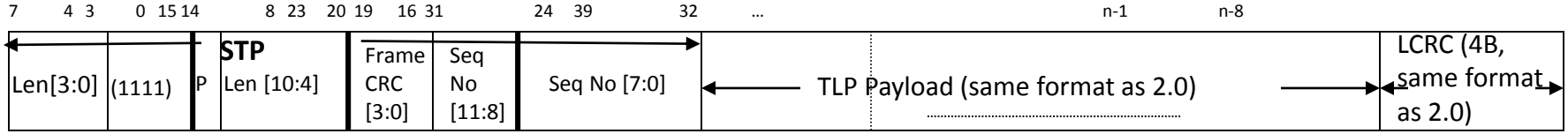
- **128b/130b Encoding similar between 8.0 GT/s and 16.0 GT/s**
  - Minor enhancements for 16.0 GT/s; Triple bit flip detection guarantee maintained
- **Lane Level Encoding: 2 bit Sync header, 128 bit payload**
  - Two types of Blocks:
    - Data Blocks: 10b Sync Header. Used for TLP, DLLP, IDL.
    - Ordered Set Blocks: 01b Sync Header. One OS per Block.
- **Scrambling provides edge density**
  - Sync header not scrambled
  - Payload in Data Blocks always scrambled
  - Ordered Set payload not scrambled except last 15 Symbols of TS1/ TS2
  - Degree 23 polynomial ($G(X) = X^{23} + X^{21} + X^{16} + X^{8} + X^{5} + X^{2} + 1$)
    - Different taps for 8 adjacent lanes (or different seeds for same tap)
    - Minimizes cross talk as well as baseline wander
  - Electrical Idle Exit Ordered Set resets LFSR (Recovery/ Config)
- **Electrical Idle Exit Ordered Set used for Block Alignment**
  - Substitutes COM used for Symbol lock in 8b/10b
- **Framing token defines the length of packets in Data Blocks**
  - Multiple packets can exist in a Data Block
  - A packet may span across multiple Data Blocks

Data Block: Framing Tokens

- **First Symbol of token indicates packet type:**
  - 00000000: IDL; xxxx1111: STP; 11110000: SDP
  - Hamming distance 4 guarantees triple bit flip detect
- **Token length is variable and indicates location of next token**
- **IDL Token is 1 Symbol. No payload (PAD merged with IDL).**
- **SDP Token is 2 Symbols**
  - DLLP is 8 Symbols with no explicit End
- **STP Token is 4 Symbols**
  - Variable TLP Length; length field in STP protected by its own CRC/ parity



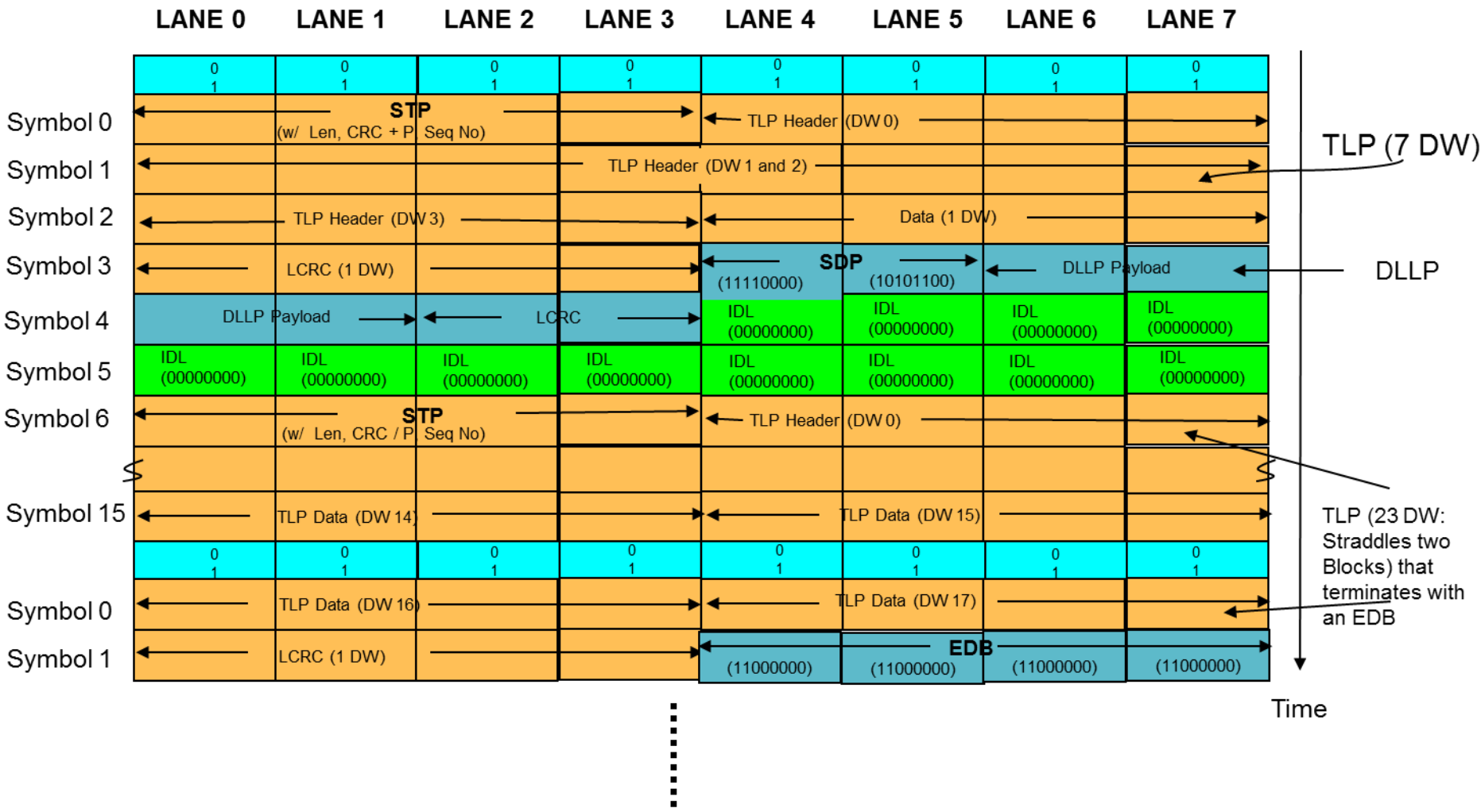
(DLLP Layout)



[Len[10:0]: length of the TLP in DWs, Frame CRC[4:0]: Check Bits covering Length[0:10], P: Frame Parity, No END]

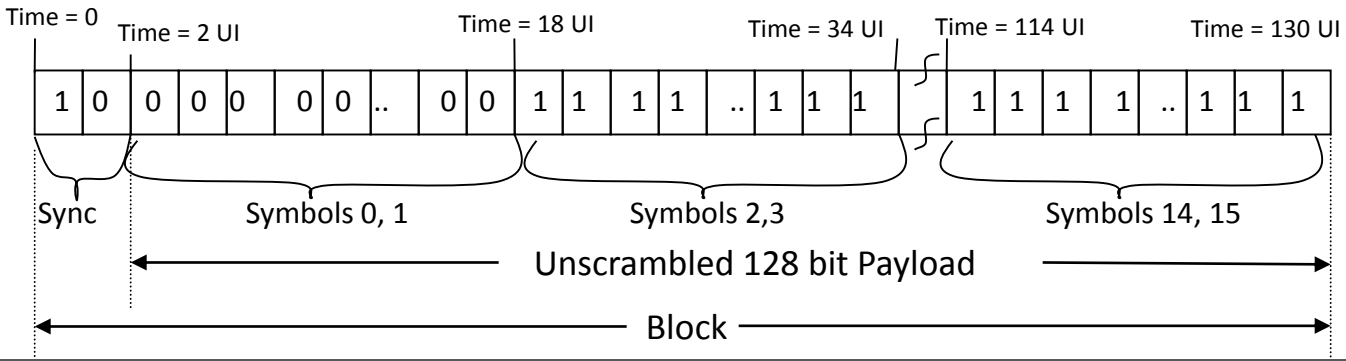
(TLP Layout)

Example Data Blocks in a x8 Link



Ordered Sets

- **First Symbol indicates the OS type**
  - Not scrambled; DC balanced; at a Hamming distance of 4 from each other
  - TS1: 1E, TS2:2D, EIEOS: 00, SKP: AA, SKP_CTL: 78, EIOS: 66, FTS: 55, SKP_END/ SDS: E1
- **None of the Ordered Sets except TS1/TS2 are scrambled. TS1/TS2:**
  - Symbol 0 not scrambled; Symbols 1-13 scrambled; Symbols 14,15 DC balance or scrambled
- **EIEOS: Low Frequency; Block Alignment, reset scrambler: Recovery/ Config**
  - 16.0 GT/s: 16 0s followed by 16 1s vs 8.0 GT/s: 8 0s followed by 8 1s
- **EDS Token is sent in the last DW of last data block prior to switching to OS**
  - Ensures a 2-bit error with the sync header does not alias to a TLP/DLLP
- **SDS (Start Data Stream) OS sent prior to first data block after a stream of OS**
  - Ensures a 2-bit error with the sync header does not alias to a TLP/DLLP



(16.0 GT/s EIEOS)

Error Detection and Recovery

- **Framing error detected by the physical layer based on some rules. Some examples:**
  - Token does not match defined types (first byte not SDP, STP, IDL, ...)
  - Sync header is 00b or 11b
  - Same sync character not present in all lanes after deskew
  - CRC / parity error in the length field of an STP token
  - No EDS in the Data Block prior to the first OS
- **Any framing error requires directing LTSSM to Recovery**
  - Stop processing any received TLP/ DLLP
  - Block lock acquired with EIEOS
  - Scrambler reset with each EIEOS
- **Error Detection Guarantees**
  - Triple bit flip detection within each TLP/ DLLP/ IDL/ OS

SKP Ordered Sets

- **Usages: Logic Analyzer, Clock Compensation, Lane Error Detection**
- **Issue with existing parity mechanism for Lane Error Detection:**
  - In a predominantly Idle Link, a bit flip will most likely result in a Framing Error
  - Link goes through Recovery => Parity information from previous L0 is lost
  - Made enhancements to deal with Framing Errors, still does not address Retimers
- **Enhancements with 16.0 GT/s Data Rate: new Control SKP OS**
  - Robust detection of Lane Error while identifying the Link Segment with Retimer(s)
  - Lane Margining at Retimer Receiver with new Control SKP OS
- **Alternate between Standard SKP OS and Control SKP OS at 16.0 GT/s**
  - Variable length at Receiver even though Transmitter sends 16 Symbols
  - Received Payload can be 8, 12, 16, 20, or 24 Symbols
  - Add or delete 4 SKP symbols in each receiver
  - Block boundary needs to be adjusted at end of the Block
- **Scrambler not advanced during transmit/ receive of SKP OS**
- **Standard SKP OS and Control SKP OS**
  - Symbols 0 through 4N-1 is AAh (SKP) same
  - Symbol 4N differentiates between SKP OS and Control SKP OS
  - Symbols 4N+1 through 4N+3 are different payloads between the two types of SKP OS

Control SKP Ordered Set at 16.0 GT/s

- **Robust Lane error detection with Link Segment identification – Data Parity from Port and each Retimer**
  - Source Port sends identical values in *Data Parity*, *First Retimer Data Parity* and *Second Retimer Data Parity*
  - Retimers overwrite their Data Parity with computed value
  - On mismatch Dest Port captures all 3 values (new registers)
  - Parity only initialized with Control SKP OS
- **Receiver Margining of Retimer**
  - Margin CRC & Margin parity protect Symbols 4N+2 and 4N+3
  - Triple bit flip detection guarantee
  - Ability to read Retimer registers (optional)

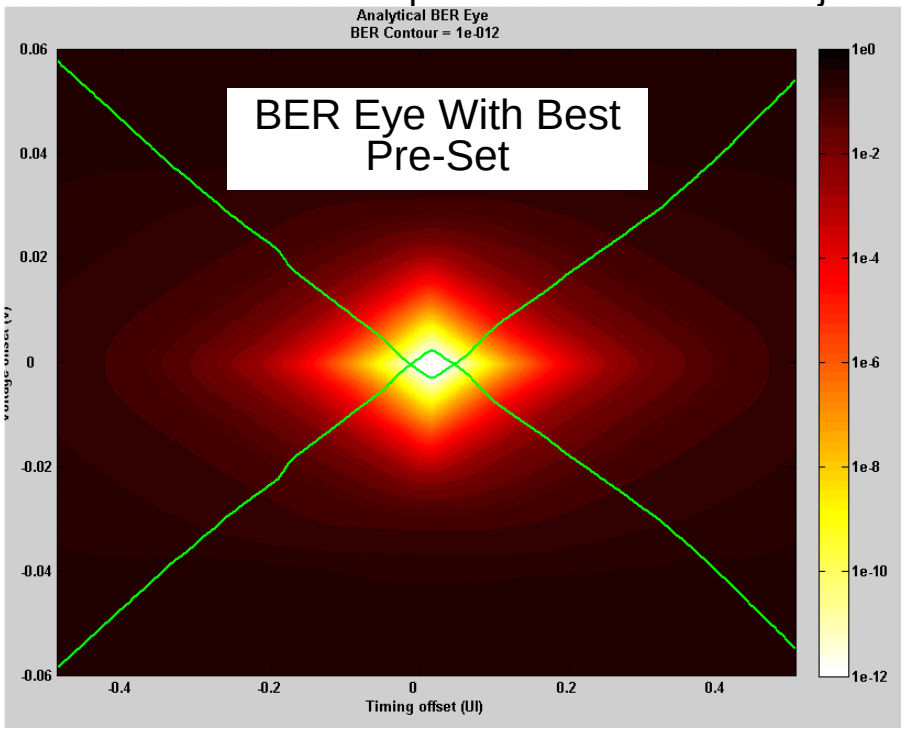
Symbol No	Value	Description
0 - (4N-1) (N = 1..5)	AAh	SKP Symbol
4N	78h	SKP_END_CTL Symbol
4N+1	00h-FFh	Bit 7: Data Parity Bit 6: First Retimer Data Parity Bit 5: Second Retimer Data Parity Bits [4:0]: Margin CRC
4N+2	00h-FFh	Bit 7: Margin Parity Bits [6:0]: Margin Command/ Status
4N+3	00h-FFh	Bits [7:0] : Margin Command/ Status

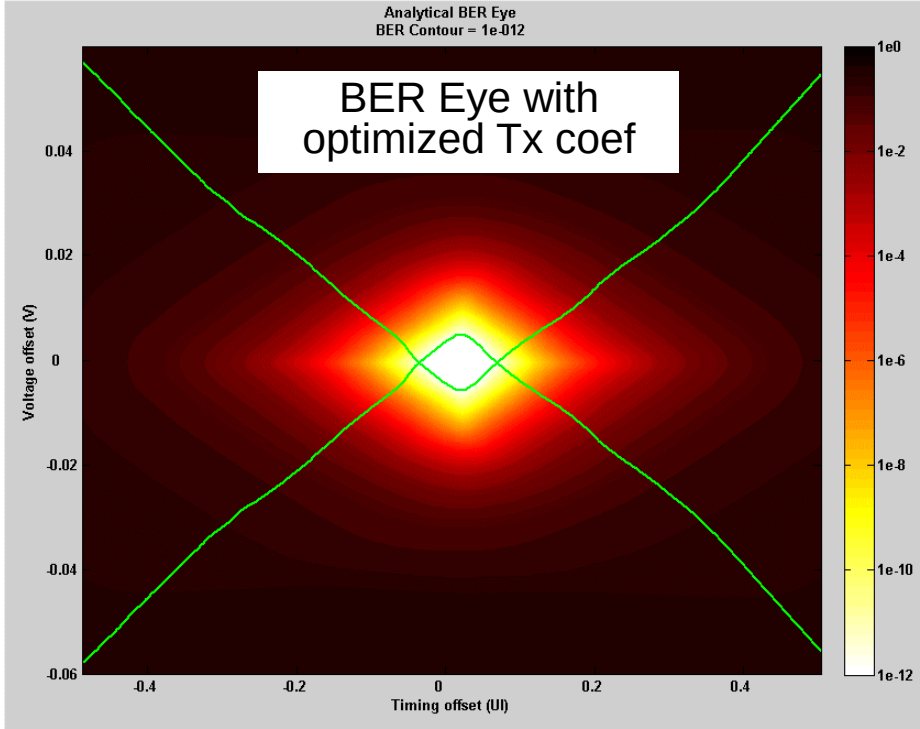
Agenda

- Background
- 128b/130b Encoding Scheme
- **Transmitter Equalization and Training**
- Testability Features and Receiver Margining
- Configuration Registers for 16.0 GT/s
- Summary

Transmitter Equalization

- **2.5 GT/s and 5.0 GT/s: Fixed de-emphasis for Link**
- **8.0 GT/s and 16.0 GT/s: Analysis demonstrates need for per Tx-Rx EQ**
  - Variations in receiver design, channel, PVT
  - Adjust each Tx by its Rx individually
  - Start with a preset value and then adjust dynamically



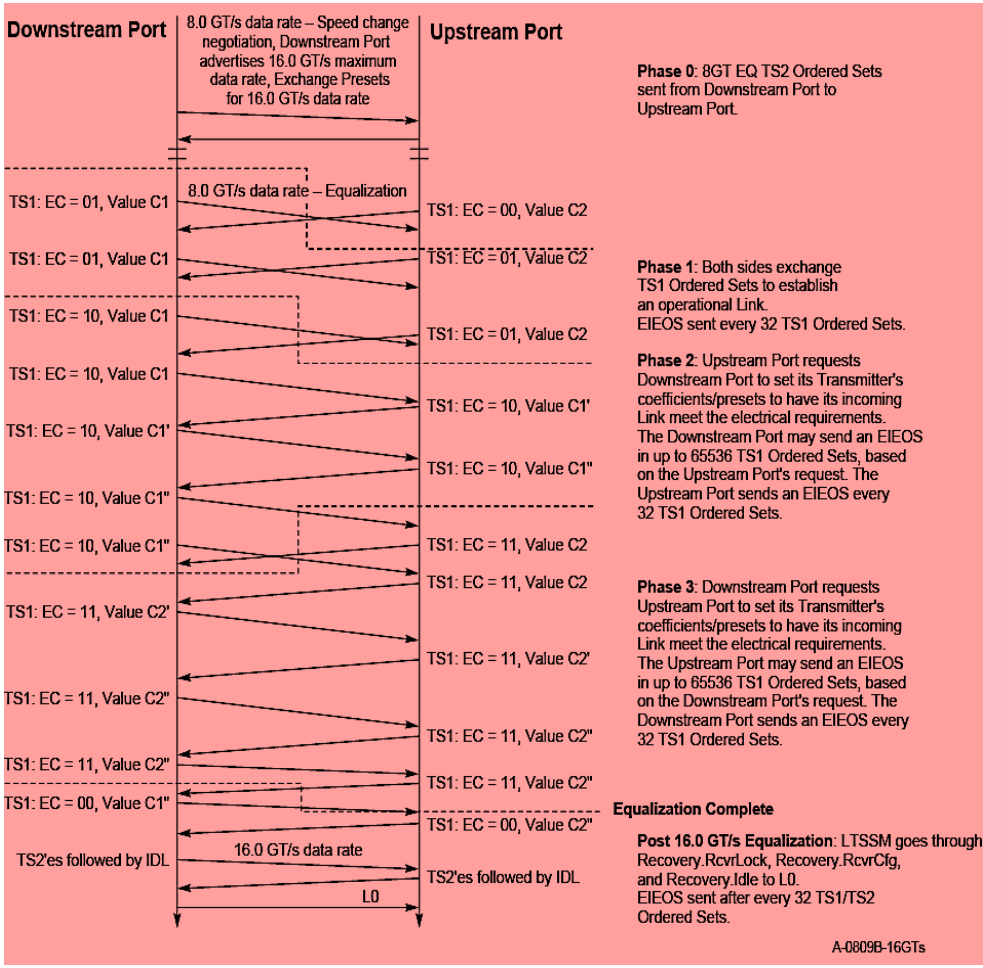


Results from an 18" 2C channel at 8.0 GT/s **Source: Intel Corporation**

Co-efficient based Tx EQ provides better margin

Equalization at 16.0 GT/s

- **16.0 GT/s EQ only after 8.0 GT/s EQ has successfully completed**
  - DSP withholds advertising 16.0 GT/s Data Rate in Recovery until 8.0 GT/s EQ has successfully completed
  - DSP responsible for L0->Recovery transition and advertising 16.0 GT/s in the autonomous EQ
  - No link-width downsizing or power management with autonomous EQ
  - DLLP exchange withheld till EQ completes for autonomous
- **8GT EQ TS2 for exchanging presets**
  - USP can request the initial preset
- **Identical presets and coefficient rules for 8.0 GT/s and 16.0 GT/s**
  - No Receiver Preset hints at 16.0 GT/s

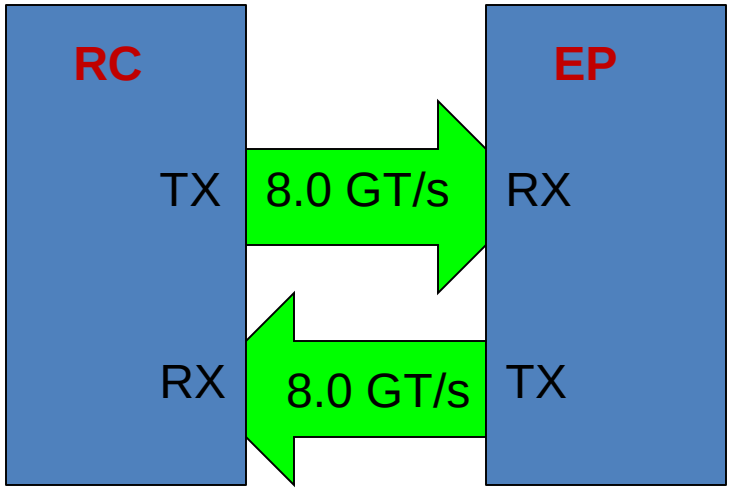


Equalization – Stage 0

Starting TxEq presets sent from Downstream Port to Upstream Port on a per Lane basis using 8GT EQ TS2 (prior to Link going to 16.0 GT/s) (optionally USP can request DSP also)

- Preset is transferred in 8GT EQ TS2 Ordered Sets
  - A Port may use a different preset in its Tx than it requests its Link Partner to use
  - Preset values (for both ports) come from the Downstream Port's (HwInit) CSR (USP's request, if implemented, is implementation specific)

Preset
0
1
2
3
4
7
8
9
10

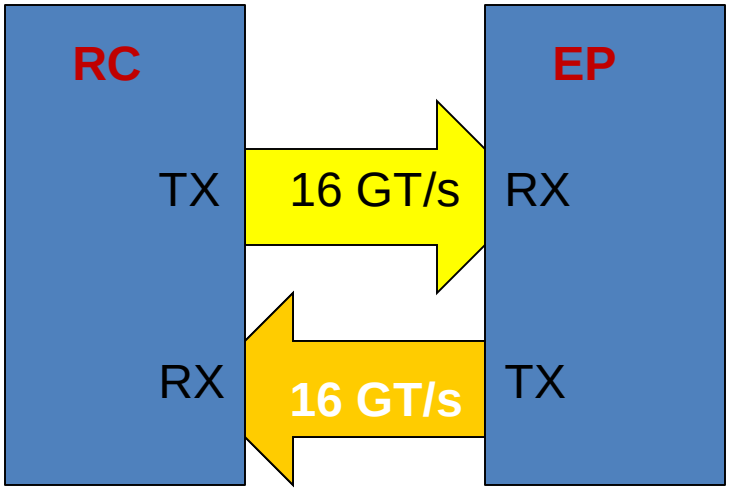


Source: Intel Corporation

Equalization – Stage 1

Starts after Link transitions to 16.0 GT/s. Both Ports use the TxEq Presets from Stage 0. Corresponds to Phase 1 in Downstream Port and Phases 0 and 1 in Upstream Port.

Expectation is that link will operate “good” enough to allow progression at 16.0 GT/s (BER <= 10^{-4}) in 24 msec; else link will go to a lower Data Rate



Source: Intel Corporation

Back Channel – Stages 2/ 3

Stage 2: Intended for Upstream Port to achieve BER <= 10^{-12}. Starts at the preset. Coefficients/ presets are exchanged in sub-loops until this is accomplished within 24 ms
A Port may decide not to make any new requests. Corresponds to Phase 2

- Receiver full swing (FS) defines granularity of coeff
  - Table at bottom-right is for illustrative purposes
  - X-axis is pre-cursor, y-axis post-cursor, diagonal defines the boostline
  - Each tile represents a coeff (e.g. p7=4/6, p8=5/5, etc)
  - Numbers in tiles represent presets; black tiles are illegal coeff space

Example: start from preset 7 (coef=4/6)

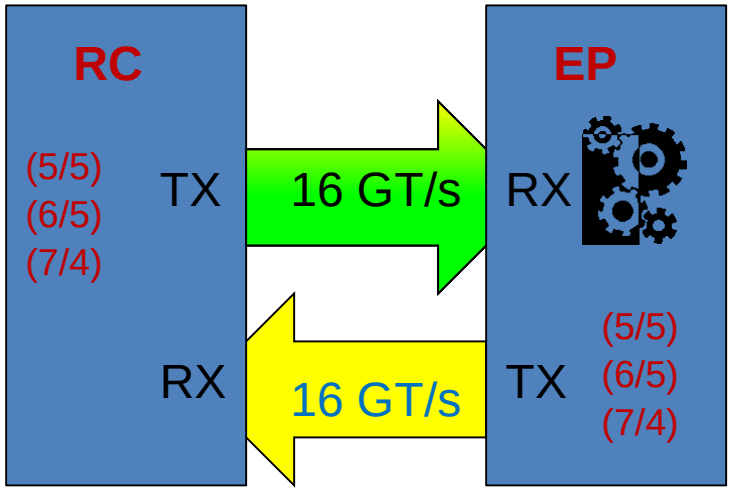
1st sub-loop

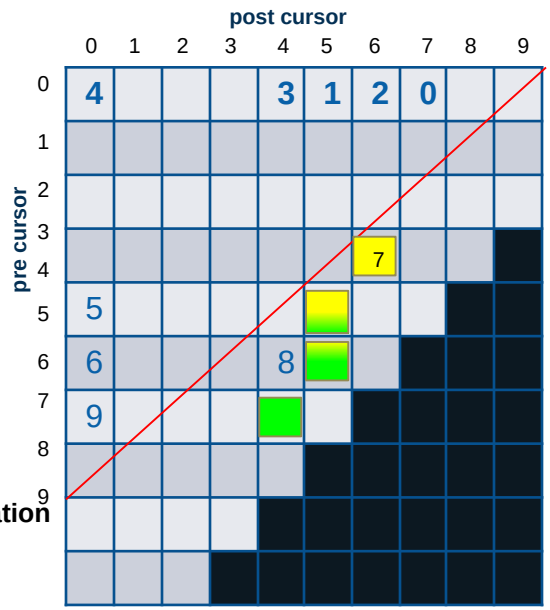
a. EP Rx eval reveals need for less post, more pre
b. EP sends (5/5) to RC
c. RC applies (5/5) to TX
d. RC echo's (5/5) to EP

2nd sub-loop

a. EP Rx eval needs more pre, post ok
b.→ d. repeat with (6/5)

3rd sub-loop finds good result with (7/4) so moves to phase 3





Source: Intel Corporation

Stage 3/ Phase 3 is same as phase 2 in opposite direction
Downstream Port may skip Phase 2/ 3 if presets are good enough for Link
Retimers perform independent EQ but within the Phases of the Ports
Retimer Enhancements for 16.0 GT/s : "Retimer Equalization Extend" to complete Ph 2/ 3

Equalization Procedure

- **Expected to be done once autonomously after Link trains to L0**
  - No DLLP/TLP exchange till equalization completes
  - Ensures no TLP timeout as equalization can take more than 100 ms
  - Software polls DL_Active prior to accessing downstream component
- **Software can perform EQ by accessing CSRs in Downstream Port**
  - Must ensure no side-effects (e.g., no timeout)
- **A device may withhold 8.0 GT/s or 16.0 GT/s Data Rate (and EQ)**
  - If its associated software can guarantee no side effects of doing equalization when it advertises 8.0 GT/s or 16.0 GT/s Data Rate
- **Error during equalization or later**
  - Not expected to redo equalization except error condition
  - Downstream Port can redo EQ
  - Upstream Port must report in its register and request
  - Downstream Port has two choices: (i) redo equalization (ii) log and report

Agenda

- Background
- 128b/130b Encoding Scheme
- Transmitter Equalization and Training
- **Testability Features and Receiver Margin**
- Configuration Registers for 16.0 GT/s
- Summary

Compliance Patterns

- **Same entry/exit mechanisms as 3.0: CSR, TS Ordered Set, CLB/CBB**
- **Preset used during compliance**
  - CSR based: Link Control 2 Register[15:12]
  - TS Ordered set: Symbol 6 of received TS1 OS
  - CLB/CBB: Cycle through 2.5GT/s, 5.0GT/s at -3.5 dB, 5.0GT/s at -6 dB, 11 presets at 8.0GT/s, 11 presets at 16.0GT/s, Settings #26 - #34 at P4 (no EQ) at 16.0 GT/s to measure Tx jitter
    - Moves by detecting exit from electrical idle
    - Setting #26: Jitter measurement pattern in all Lanes
    - Setting #27: Jitter measurement pattern in Lanes 0, 8, ..; compliance pattern in other Lanes
    - Setting #28: Jitter measurement pattern in Lanes 1, 9, ..; compliance pattern in other Lanes **……..**
    - Setting #34: Jitter measurement pattern in Lanes 7, 15, ..; compliance pattern in other Lanes
- **Compliance Pattern: 36 Blocks**
  - Sync Hdr: 01b. Payload: 64 1's followed by 64 0's
  - 2 blocks with Sync Hdr: 01b, Payload: different values in different Lanes to achieve DC balance across 36 blocks
  - EIEOS (scrambler gets reset)
  - 32 Data Blocks: Payload scrambled 16 IDL Symbols (different seed/ taps in 8 adjacent Lanes)
- **Jitter Measurement Pattern: Sync hdr 01b, 16 Symbols of 55h unscrambled (Clk pattern)**

Loopback and Modified Compliance Pattern

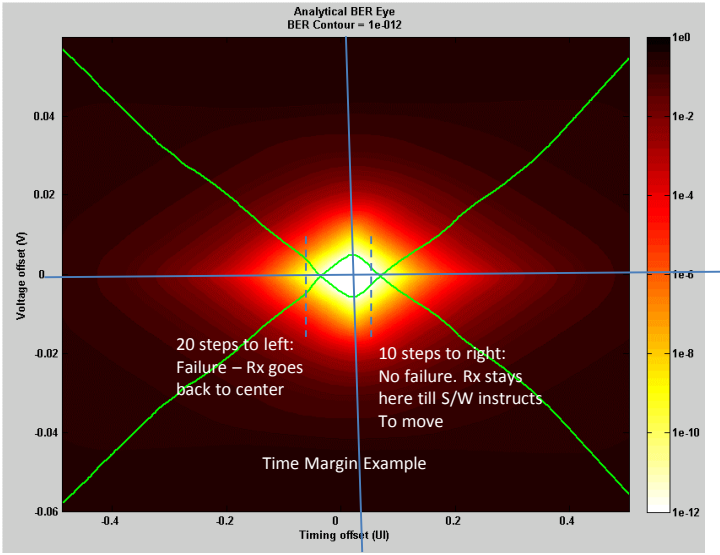
- **Modified Compliance pattern: Same 65792 Blocks**
  - EIEOS, 256 Data Blocks, 255 Sets of {SKP OS, 256 Data Blocks}
  - Presets chosen using the same mechanism as in compliance pattern (CSR or TS1 based)

- **Loopback mechanisms same as PCIe 3.x**
  - 128b/130b used: Either 01b or 10b sync header must be used
    - LB Master should avoid using SKP OS, EIEOS, and EIOS as the payload to compare after loopback, since they have specific purposes in Loopback
  - SKP OS sent periodically for Slave to adjust for ppm differences
  - Same LTSSM transitions for Loopback
    - Need to send EIEOS on LB Entry – once every 32 blocks
    - Slave may switch to loopback data at any arbitrary boundary
    - Master re-acquires block alignment in LB.Entry after the Slave loops back

Lane Margining at Receiver

- **Problem Statement: No standard way to margin real systems**
  - Rx CEM Compliance testing expensive and tests each device/ Lane individually
  - Does not test in L0 (no EQ handshake) or in a system or PVT variations
  - Receivers may behave differently in test mode (Rx compliance, Loopback) vs L0
  - No way to know how much margin exists in a given Link in a system
  - Retimers pose additional challenges

- **Solution: Non-destructive margining during L0**
  - No test equipment needed
  - Architected CSRs to control and report margin
  - Retimers accessed through Control SKP OS
  - Allow time for software help in set-up
  - Flexibility for different implementations

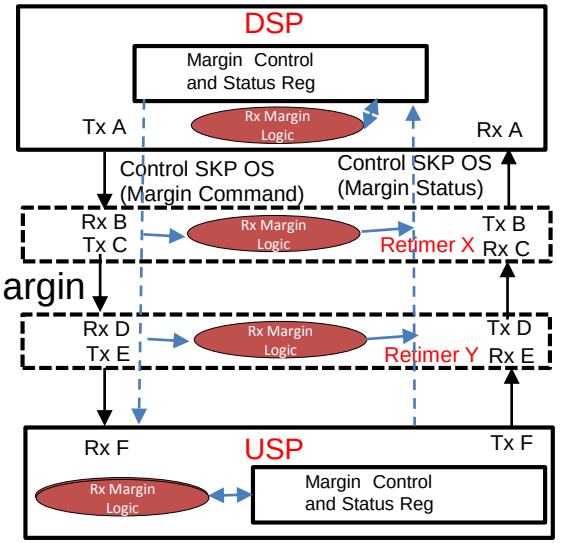


Source: Intel Corporation

- **Possible Usage Examples:**
  - Platform qual (across wide range of platforms, components, PVT variations), Debug, Run-time diagnostics, Checking margin after hot-plug, etc.

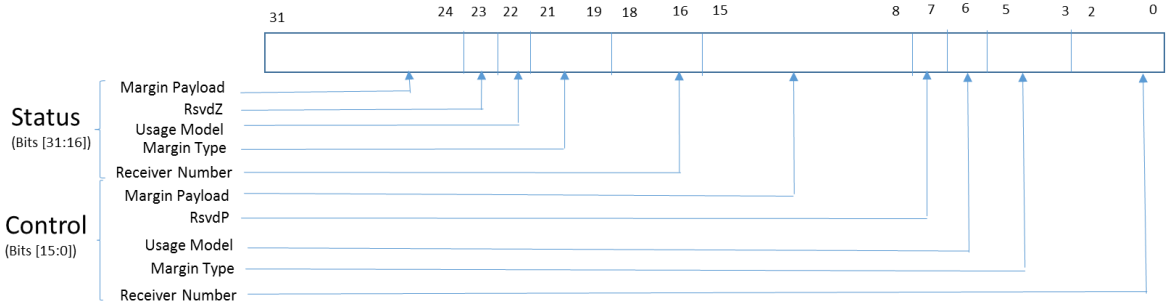
Lane Margin Control and Status Register and Control SKP OS

- **Lane Margin Control and Status Register present in USP and DSP (per Lane) - separate Cap structure**
  - USP and DSP margined only through their respective CSR
- **Retimers margined using DSP CSR via Control SKP OS**
  - Margin commands flow downstream
  - Margin status flows upstream
  - Identical bits between Control SKP OS and the DSP CSR
  - Control SKP OS Symbols 4N+2 and 4N+3 ignored if Margin Parity/ Margin CRC checks fail
- **Usage Model: 0b => Rx margining, 1b: Future use**
- **Margin Command/ Status: Combination of Margin Type and Payload**
- **Receiver Number: Target Rx command/ response**
  - 000b: Broadcast, 001b: Rx A, 010b: Rx B, 011b: Rx C, 100b: Rx D, 101b: Rx E, 110b: Rx F, 111b: Reserved



Symbol	Fields
4N+2	Bit 6: Usage Model (0b: Lane Margining at Receiver, 1b: Reserved) Bits [5:3]: Margin Type Bits [2:0]: Receiver Number
4N+3	Bits [7:0] : Margin Payload

(Control SKP OS)



(Lane Margin Control and Status Register)

Margin Commands and Responses

Command				Response	
Margin Command	Margin Type [2:0]	Valid Rx No(s) [2:0]	Margin Payload [7:0]	Margin Type [2:0]	Margin Payload [7:0]
No Command	111b	000b	9Ch (No Command is an independent command sent by the USP – acts as a response)		
Read Retimer Register (Optional)	001b	010b, 100b	Reg. Offset in bytes: 00h-87h, A0-FFh	001b	Register value, if supported; Else 00h
Report Margin Parameters (different sets)	001b	001b-110b	88h-90h	001b	Response for parameter requested; e.g., for 89h: {Margin Payload [7] = Reserved Margin Payload[6:0] = $M_{NumVoltageSteps}$}
Set Error Count Limit	010b	001b-110b	Margin Payload [7:6] = 11b Margin Payload[5:0] = Error Count Limit	010b	Margin Payload [7:6] = 11b Margin Payload[5:0] = Error Count Limit registered by the target Receiver
Go to Normal Settings / Clear Error Log	010b	000b-110b	0Fh/ 55h	010b	0Fh/ 55h
Step Margin to timing offset to right/left of default	011b	001b - 110b	Margin amount	011b	Margin Payload [7:6] = {11b- NAK (e.g., timing > 0.2UI), 10b – Margin in progress, 01b – Set up in progress (valid within 100 ms of cmd), 00b – too many errors, Rx reverted to normal settings} Margin Payload[5:0] = $M_{ErrorCount}$
Step Margin to voltage offset to up/down of default	100b	001b - 110b	Margin amount	100b	Margin Payload [7:6] = {11b – NAK, 10b – Margin in progress, 01b – Set up in progress (valid within 100 ms of cmd), 00b – too many errors, Rx reverted to normal settings} Margin Payload[5:0] = $M_{ErrorCount}$
Vendor Defined	101b	001b-101b	Vendor Defined	101b	Vendor Defined

Margin Command and Response Flow

- **Only at 16.0 GT/s in L0 with all margined Lanes active (no ASPM on)**
- **For a margin command (through CSR for DSP/ USP or Downstream Control SKP OS), each receiver checks if it is the target of the command**
  - e.g., Receiver number matches, including broadcast and have a valid command combination
  - USP and DSP respond directly on their CSR within 1 msec
  - Retimers report through upstream Control SKP OS (1 msec) which then is logged in the DSP CSR
  - Broadcast commands including Retimers: Response always logged from farthest Retimer
  - With CRC/parity and constant resend of command/ response, we have a guaranteed delivery mechanism
- **Once received, a `Step Margin' command is in effect in L0 until:**
  - it receives a `Go to Normal Settings' or
  - a different `Step Margin' command or
  - Link enters a state other than L0 or Recovery or undergoes a speed change or
  - for non-independent sampler (and optionally for independent error samplers) if errors exceed `Error Count Limit'
    - Receiver must return to normal settings and report status as "too many errors" (for `Margin Status Execution Status')
  - Reports Status: NAK, Set up in progress, Margining in progress (with error count), too many errors (w/ error count)

Margin Command and Response Flow (contd)

- **Margined receiver tracks errors as follows:**
  - Non-Independent Error Sampler: Data parity mismatch in L0 or entry to Recovery
  - Independent Error Sampler: bit mismatch in L0 (sample count must be consistent with bit matching)
- **For non-independent sampler: If LTSSM enters Recovery, must apply the offset within 128 usec of re-entry to L0**
- **Target Receiver clears error count on receipt of "clear error log" cmd**
  - Enables S/W to clear errors without interrupting margining for longer runs
- **Software reads the capabilities of a Receiver prior to margining**

  e.g.: $M_{IndErrorSampler}$, $M_{SampleReportingMethod}$, $M_{IndLeftRightTiming}$, $M_{IndUpDownVoltage}$, $M_{VoltageSupported}$, $M_{numVoltageSteps}$ ….
- **Software must broadcast `No Command', check it completed prior to issuing a new margin command**
- **At the end of margining in a given direction (voltage/ timing and up/down/left/right), software must broadcast `Go to Normal Settings', `No Command', `Clear Error Log', and `No Command' in series in the Downstream and Upstream Ports, after ensuring each command has been acknowledged by the target Receiver.**

Agenda

- Background
- 128b/130b Encoding Scheme
- Transmitter Equalization and Training
- Testability Features and Receiver Margin
- **Configuration Registers for 16.0 GT/s**
- Summary

16.0 GT/s Related Configuration Registers

- **16.0 GT/s Data Rate reflected in existing registers**
  - E.g., `Supported Link Speeds Vector' in `Link Capabilities 2 Register'
- **Physical Layer 16.0 GT/s Extended Capability structure**
  - 16.0 GT/s Status Register: (0Ch) 5 bits representing EQ
    - {Link EQ req, Ph 3 / 2/ 1 successful, EQ successful}
  - Local Data Parity Mismatch (10h) (one bit per Lane)
  - First and Second Retimer Parity Mismatch (14h and 18h)
  - Lane Equalization Control from offset 20h onwards; 1 Byte per Lane, DW granularity
    - Bits [3:0]: Downstream Port 16.0 Transmitter Preset
    - Bits [7:4]: Upstream Port 16.0 Transmitter Preset
- **Retimer Related:**
  - `Retimer Presence Detect Supported' and `Two Retimers Presence Detect Supported' in `Link Capabilities 2 Register'
  - `Two Retimers Presence Detected' in addition to `Retimer Presence Detected' in `Link Status 2 Register'
- **Physical Layer 16.0 GT/s Margining Extended Capability structure**
- **`Cross-Link Resolution' (2 bits) added to `Link Status 2 Register'**
  - S/W needs to comprehend for various usages (e.g., margining)

Agenda

- Background
- 128b/130b Encoding Scheme
- Transmitter Equalization and Training
- Testability Features and Receiver Margin
- Configuration Registers for 16.0 GT/s
- **Summary**

Summary

- **PCIe 4.0 Base Spec Rev 0.9 in publication/ review process**
- **Introduces a new Data Rate at 16.0 GT/s**
  - Doubled bandwidth again – now at 4th generation!
  - Expect Retimers for longer channel reach
- **Leverage existing Encoding, Tx Equalization, and compliance mechanisms**
- **Enhancements include Retimer support, Fault isolation, compliance testing, and run-time margining during L0 in platforms**
- **Track the PCIe 4.0 Base Spec and provide your feedback**

Thank you for attending the PCI-SIG Developers Conference 2017.

For more information, please go to www.pcisig.com

Don't forget to submit your feedback via the mobile app!

Download the app by searching stores for **PCI-SIG Developers Conference** or entering the following URL into your mobile browser: **https://crowd.cc/s/EOtK**

Enter event code: **DevCon17**

Alternatively, access here: https://event.crowdcompass.com/devcon-17

Note: Create an account within the app so Admin knows who to contact if selected as the prize winner.

You will earn 1 raffle entry for each session you provide feedback on (up to 11 sessions).
General survey feedback = 1 raffle entry.